

What is claimed is:

1: 1. An instruction scheduling method comprising:

2 a priority calculation step of calculating a priority
3 of each of a plurality of instructions that are subjected
4 to scheduling, based on dependencies between the plurality
5 of instructions and constraints of hardware resources for
6 processing the plurality of instructions, the dependencies
7 being data dependency, anti-dependency, and output
8 dependency; and

9 an execution timing decision step of deciding an
10 execution timing of an instruction having a highest
11 priority.

1 2. The instruction scheduling method of Claim 1,

2 wherein the priority calculation step includes:

3 a precedence constraint rank calculation substep of
4 calculating a precedence constraint rank of each of the
5 plurality of instructions, wherein (a) if the instruction
6 has a succeeding instruction which is anti-dependent or
7 output dependent on the instruction, the precedence
8 constraint rank of the instruction is equal to a precedence
9 constraint rank of the succeeding instruction, and (b)
10 if the instruction has a succeeding instruction which is
11 data dependent on the instruction, the precedence
12 constraint rank of the instruction is higher than a

13 precedence constraint rank of the succeeding instruction;
14: and

15 a resource constraint evaluation substep of judging
16 (i) whether the instruction has a succeeding instruction
17 which is dependent on the instruction, (ii) whether the
18 instruction and the succeeding instruction have an equal
19 precedence constraint rank, and (iii) whether a hardware
20 resource for processing the instruction cannot process
21 the instruction and the succeeding instruction in parallel,
22 and

23 the priority calculation step raises the precedence
24 constraint rank of the instruction and sets the raised
25 precedence constraint rank as a priority of the instruction
26 if all of the judgments (i), (ii), and (iii) are in the
27 affirmative, and sets the precedence constraint rank of
28 the instruction as the priority of the instruction if any
29 of the judgments (i), (ii), and (iii) is in the negative.

1 3. The instruction scheduling method of Claim 1,

2 wherein the priority calculation step includes:

3 a precedence constraint rank calculation substep of
4 calculating a precedence constraint rank of each of the
5 plurality of instructions, wherein (a) if the instruction
6 has no succeeding instruction which is dependent on the
7 instruction, the precedence constraint rank of the

8 instruction is 1, (b) if the instruction has one or more
9 succeeding instructions which are anti-dependent or output
10 dependent on the instruction, the precedence constraint
11 rank of the instruction is a highest one of precedence
12 constraint ranks of the succeeding instructions, and (c)
13 if the instruction has one or more succeeding instructions
14 which are data dependent on the instruction, the precedence
15 constraint rank of the instruction is a sum of 1 and a
16 highest one of precedence constraint ranks of the
17 succeeding instructions; and
18 a resource constraint evaluation substep of
19 calculating a resource constraint value of the instruction,
20 by dividing a total number of instructions which are to
21 be processed by a hardware resource for processing the
22 instruction and whose execution timings have not been
23 decided, by a maximum number of instructions that can be
24 processed in parallel by the hardware resource, and
25 the priority calculation step sets the resource
26 constraint value as a priority of the instruction if the
27 resource constraint value is larger than the precedence
28 constraint rank, and sets the precedence constraint rank
29 as the priority of the instruction if the resource
30 constraint value is no larger than the precedence
31 constraint rank.

1 4. An instruction scheduling method for sequentially
2 deciding execution timings of instructions that are
3 subjected to scheduling, comprising:
4 a decision judgment step of judging, after an
5 execution timing of a first instruction is decided, whether
6 an execution timing of a second instruction can be decided
7 so as to be within a predetermined time period, based on
8 a constraint of a hardware resource for processing the
9 second instruction; and
10 a redecision step of retracting, if the judgment is
11 in the negative, the decision of the execution timing of
12 the first instruction and deciding an execution timing
13 of an instruction other than the first instruction.

1 5. The instruction scheduling method of Claim 4,
2 wherein the predetermined time period is expressed
3 by a number of clock cycles,
4 the decision judgment step includes:
5 a resource constraint evaluation substep of
6 calculating a resource constraint value of the second
7 instruction, by dividing a total number of instructions
8 which are to be processed by the hardware resource and
9 whose execution timings have not been decided, by a maximum
10 number of instructions that can be processed in parallel
11 by the hardware resource, and

12 the decision judgment step judges in the negative
13: if the resource constraint value is larger than the number
14 of clock cycles.

1 6. A program conversion method characterized in that:
2 an input program is converted to an object program
3 including a plurality of instructions, and an execution
4 timing of each of the plurality of instructions in the
5 object program is decided using the instruction scheduling
6 method of one of Claims 1 to 5.

1 7. An instruction scheduling device comprising:
2 a priority calculation unit operable to calculate
3 a priority of each of a plurality of instructions that
4 are subjected to scheduling, based on dependencies between
5 the plurality of instructions and constraints of hardware
6 resources for processing the plurality of instructions,
7 the dependencies being data dependency, anti-dependency,
8 and output dependency; and
9 an execution timing decision unit operable to decide
10 an execution timing of an instruction having a highest
11 priority.

1 8. An instruction scheduling device for sequentially
2 deciding execution timings of instructions that are

3 subjected to scheduling, comprising:
4 a decision judgment unit operable to judge, after
5 an execution timing of a first instruction is decided,
6 whether an execution timing of a second instruction can
7 be decided so as to be within a predetermined time period,
8 based on a constraint of a hardware resource for processing
9 the second instruction; and
10 a redecision unit operable to retract, if the judgment
11 is in the negative, the decision of the execution timing
12 of the first instruction and decide an execution timing
13 of an instruction other than the first instruction.

1 9. A computer-executable program for instruction
2 scheduling, having a computer execute:

3 a priority calculation step of calculating a priority
4 of each of a plurality of instructions that are subjected
5 to scheduling, based on dependencies between the plurality
6 of instructions and constraints of hardware resources for
7 processing the plurality of instructions, the dependencies
8 being data dependency, anti-dependency, and output
9 dependency; and

10 an execution timing decision step of deciding an
11 execution timing of an instruction having a highest
12 priority.

1 10. A computer-executable program for sequentially
2 deciding execution timings of instructions that are
3 subjected to scheduling, having a computer execute:
4 a decision judgment step of judging, after an
5 execution timing of a first instruction is decided, whether
6 an execution timing of a second instruction can be decided
7 so as to be within a predetermined time period, based on
8 a constraint of a hardware resource for processing the
9 second instruction; and
10 a redecision step of retracting, if the judgment is
11 in the negative, the decision of the execution timing of
12 the first instruction and deciding an execution timing
13 of an instruction other than the first instruction.

1 11. A computer-readable storage medium storing the program
2 of one of Claims 9 and 10.